AMENDMENTS TO THE CLAIMS

1. (Currently amended): A method for booting a non-uniform-memory-access (NUMA) machine, comprising:

prior to beginning booting of any of a plurality of standalone, symmetrical multiprocessing (SMP) systems, configuring each one of said [[a]] plurality of standalone, symmetrical multiprocessing systems to operate within a NUMA system;

prior to beginning booting of any of said plurality of SMP systems, assigning a NUMA identification to each of the multiprocessing systems, wherein each identification is unique; and

concurrently booting all of the multiprocessing systems together in NUMA mode in one-pass, ~~wherein~~ memory coherency being [[is]] established prior to ~~at the beginning of~~ the execution, by any of said plurality of SMP systems, of [[the]] system firmware.

2. (Original): The method according to claim 1, wherein the step of configuring the multiprocessing systems further comprises:

configuring and testing host processors and memory;

configuring and testing NUMA memory;

loading firmware image into local memory and informing a hardware system console of the firmware version;

receiving a confirmation from the hardware system console that the firmware version is the same for all multiprocessing systems in the NUMA system;

configuring NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors; and

releasing all host processors to execute system firmware.

3. (Currently amended): The method according to claim 1, further comprising:

prior to beginning booting of any of said plurality of SMP systems, all of a plurality of processors within each one of said SMP systems competing with each other to become a nodal master processor;

prior to beginning booting of any of said plurality of SMP systems, designating one of said plurality of processors within each one of said SMP systems that won said competition as said nodal master processor and designating all remaining ones of said plurality of processors within each one of said SMP systems as slave processors; ~~selecting a nodal master processor within each multiprocessing system and designating all other processors within each system as nodal slave processors; and~~

prior to beginning booting of any of said plurality of SMP systems, each one of said nodal master processors competing with each other to become a NUMA master processor;

prior to beginning booting of any of said plurality of SMP systems, designating one of said nodal master processors that won said competition as said NUMA master processor and designating all remaining nodal master processors as slave NUMA processors; and ~~selecting a NUMA master processor from among the separate nodal master processors and designating all other nodal master processors as NUMA slave processor.~~

concurrently booting all of the SMP systems together in NUMA mode in one-pass utilizing said NUMA master processor.

4. (Currently amended): The method according to claim 3, further comprising:

performing one-to-one handshaking between the nodal master processor and each nodal slave processor within a multiprocessing system, wherein the handshaking synchronizes the time base register of all nodal slave processors with [[the]] a nodal time base source; and

switching the nodal slave processors in each multiprocessing system to a hypervisor environment in which the nodal slave processors become NUMA slave processors.

5. (Original): The method according to claim 3, wherein the NUMA master processor:

loads a single operating system into NUMA system memory;

transfers control to the operating system; and

runs operating system code which assigns all slave processors to a designated place within the NUMA operating system.

6. (Currently amended): A system for booting a non-uniform-memory-access (NUMA) machine, comprising:

a plurality of hardware-configuring components which, prior to beginning booting of any of a plurality of standalone, symmetrical multiprocessing (SMP) systems, configure [[a]] each one of said plurality of standalone, symmetrical multiprocessing (SMP) systems to operate within a NUMA system;

an identification component which, prior to beginning booting of any of said plurality of SMP systems, assigns a NUMA identification to each of the multiprocessing systems, wherein each identification is unique; and

a booting mechanism which concurrently boots all of the multiprocessing systems together in NUMA mode in one-pass, ~~wherein~~ memory coherency being [[is]] established prior to ~~at the beginning of~~ the execution, by any of said plurality of SMP systems, of [[the]] system firmware.

7. (Original): The system according to claim 6, wherein each hardware-configuring component further comprises:

a host testing component which configures and tests host processors and memory;

a NUMA testing component which configures and tests NUMA memory;

a software loading mechanism which loads a firmware image into local memory and informs a hardware system console of the firmware version;

a receiving component which receives a confirmation from the hardware system console that the firmware version is the same for all multiprocessing systems in the NUMA system;

an adapter-configuring component which configures NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors; and

a releasing mechanism which releases all host processors to execute system firmware.

8. (Currently amended): The system according to claim 6, further comprising:

all of a plurality of processors within each one of said SMP systems competing with each other, prior to beginning booting of any of said plurality of SMP systems, to become a nodal master processor;

one of said plurality of processors within each one of said SMP systems that won said competition being designated as said nodal master processor and all remaining ones of said plurality of processors within each one of said SMP systems being designated as slave processors prior to beginning booting of any of said plurality of SMP systems;

each one of said nodal master processors competing with each other, prior to beginning booting of any of said plurality of SMP systems, to become a NUMA master processor;

one of said nodal master processors that won said competition being designated as said NUMA master processor and all remaining nodal master processors being designated as slave NUMA processors prior to beginning booting of any of said plurality of SMP systems; and

all of the SMP systems being concurrently booted together in NUMA mode in one-pass utilizing said NUMA master processor.

~~a plurality of nodal selection mechanisms which select a nodal master processor within each multiprocessing system and designate all other processors within each system as nodal slave processors; and~~

~~a NUMA selection mechanism which selects a NUMA master processor from among the separate nodal master processors and designates all other nodal master processors as NUMA slave processors.~~

9. (Currently amended): The system according to claim 8, further comprising:

a handshaking mechanism which performs one-to-one handshaking between the nodal master processor and each nodal slave [[salve]] processor within a multiprocessing system, wherein the handshaking synchronizes the time base register of all nodal slave processors with a [[the]] nodal time base source; and

a switching mechanism which switches the nodal slave processors in each multiprocessing system to a hyper-visor environment in which the nodal slave processors become NUMA slave processors.

10. (Original): The system according to claim 8, wherein the NUMA master processor:

loads a single operating system into NUMA system memory;

transfers control to the operating system; and

runs operating system code which assigns all slave processors to a designated place within the NUMA operating system.

11. (Currently amended): A computer program product in a computer readable medium for use in a data processing system, for booting a non-uniform-memory-access (NUMA) machine, the computer program product comprising:

instructions for, prior to beginning booting of any of a plurality of standalone, symmetrical multiprocessing (SMP) systems, configuring each one of said [[a]] plurality of standalone, symmetrical multiprocessing systems to operate within a NUMA system;

instructions for, prior to beginning booting of any of said plurality of (SMP) systems, assigning a NUMA identification to each of the multiprocessing systems, wherein each identification is unique; and

instructions for concurrently booting all of the multiprocessing systems together in NUMA mode in one-pass, ~~wherein~~ memory coherency being [[is]] established ~~at the beginning of~~ the execution, by any of said plurality of SMP systems, of [[the]] system firmware.

12. (Original): The computer program product according to claim 11, wherein the step of configuring the multiprocessing systems further comprises:

instructions for configuring and testing host processors and memory;

instructions for configuring and testing NUMA memory;

instructions for loading firmware image into local memory and informing a hardware system console of the firmware version;

instructions for receiving a confirmation from the hardware system console that the firmware version is the same for all multiprocessing systems in the NUMA system;

instructions for configuring NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors; and

instructions for releasing all host processors to execute system firmware.

13. (Currently amended): The computer program product according to claim 11, further comprising:

prior to beginning booting of any of said plurality of SMP systems, all of a plurality of processors within each one of said SMP systems competing with each other to become a nodal master processor;

instructions for designating, prior to beginning booting of any of said plurality of SMP systems, one of said plurality of processors within each one of said SMP systems that won said competition as said nodal master processor and designating all remaining ones of said plurality of processors within each one of said SMP systems as slave processors;

prior to beginning booting of any of said plurality of SMP systems, each one of said nodal master processors competing with each other to become a NUMA master processor;

instructions for designating, prior to beginning booting of any of said plurality of SMP systems, one of said nodal master processors that won said competition as said NUMA master processor and designating all remaining nodal master processors as slave NUMA processors; and ~~selecting a NUMA master processor from among the separate nodal master processors and designating all other nodal master processors as NUMA slave processor.~~

instructions for concurrently booting all of the SMP systems together in NUMA mode in one-pass utilizing said NUMA master processor.

~~instructions for selecting a nodal master processor within each multiprocessing system and designating all other processors within each system as nodal slave processors; and~~

~~instructions for selecting a NUMA master processor from among the separate nodal master processors and designating all other nodal master processors as NUMA slave processors.~~

14. (Currently amended): The computer program product according to claim 13, further comprising:

instructions for performing one-to-one handshaking between the nodal master processor and each nodal slave [[salve]] processor within a multiprocessing system, wherein the handshaking synchronizes the time base register of all nodal slave processors with a [[the]] nodal time base source; and

instructions for switching the nodal slave processors in each multiprocessing system to a hyper-visor environment in which the nodal slave processors become NUMA slave processors.

15. (Original): The computer program product according to claim 13, wherein the NUMA master processor:

loads a single operating system into NUMA system memory;

transfers control to the operating system; and

runs operating system code which assigns all slave processors to a designated place within the NUMA operating system.

16. (Currently amended): A system for booting a non-uniform-memory-access (NUMA) machine, comprising:

a plurality of hardware-configuring components which, prior to beginning booting of any of a plurality of standalone, symmetrical multiprocessing (SMP) systems, configure each one of said plurality of SMP systems to operate within a NUMA system;

an identification component which, prior to beginning booting of any of said plurality of SMP systems, assigns a NUMA identification to each of the multiprocessing systems, wherein each identification is unique;

a booting mechanism which concurrently boots all of the multiprocessing systems together in NUMA mode in one-pass, memory coherency being established prior to the execution, by any of said plurality of SMP systems, of system firmware;

all of a plurality of processors within each one of said SMP systems competing with each other, prior to beginning booting of any of said plurality of SMP systems, to become a nodal master processor;

prior to beginning booting of any of said plurality of SMP systems, one of said plurality of processors within each one of said SMP systems winning said competition by acquiring a nodal semaphore, wherein only one processor can acquire and hold said nodal semaphore at a time;

said one of said plurality of processors that acquired said nodal semaphore being designated, prior to beginning booting of any of said plurality of SMP systems, as said nodal master processor and all remaining ones of said plurality of processors within each one of said SMP systems being designated as slave processors;

each one of said nodal master processors competing with each other, prior to beginning booting of any of said plurality of SMP systems, to become a NUMA master processor;

prior to beginning booting of any of said plurality of SMP systems, one of said nodal master processors winning said competition by acquiring a NUMA semaphore, wherein only one processor can acquire and hold said NUMA semaphore at a time;

said one of said nodal master processors that acquired said NUMA semaphore being designated, prior to beginning booting of any of said plurality of SMP systems, as said NUMA master processor and all remaining nodal master processors being designated as slave NUMA processors; and

all of the SMP systems being concurrently booted together in NUMA mode in one-pass utilizing said NUMA master processor.

~~a plurality of hardware-configuring components which configure a plurality of standalone, symmetrical multiprocessing systems to operate within a NUMA system;~~

~~a host testing component which configures and tests host processors and memory;~~

~~a NUMA testing component which configures and tests NUMA memory;~~

~~a software loading mechanism which loads a firmware image into local memory and informs a hardware system console of the firmware version;~~

~~a receiving component which receives a confirmation from the hardware system console that the firmware version is the same for all multiprocessing systems in the NUMA system;~~

~~an adapter configuring component which configures NUMA adapters to connect each multiprocessing system to the NUMA system and initializing all host processors;~~

~~a releasing mechanism which releases all host processors to execute system firmware;~~

~~an identification component which assigns a NUMA identification to each of the multiprocessing systems, wherein each identification is unique;~~

~~a plurality of nodal selection mechanisms which select a nodal master processor within each multiprocessing system and designate all other processors within each system as nodal slave processors;~~

~~a NUMA selection mechanism which selects a NUMA master processor from among the separate nodal master processors and designates all other nodal master processors as NUMA slave processors;~~

~~a switching mechanism which switches the nodal slave processors in each multiprocessing system to a hyper visor environment in which the nodal slave processors become NUMA slave processors; and~~

~~a booting mechanism which boots the multiprocessing systems in NUMA mode in one pass, wherein memory coherency is established at the beginning of the execution of the system firmware.~~